HEMELB ACCELERATION AND VISUALIZATION FOR CEREBRAL ANEURYSM

Sahar Soheilian Esfahani[⋆] *Xiaojun Zhai*[†] *Minsi Chen*[‡] *Abbes Amira*[⋆] *Faycal Bensaali*[⋆]
Julien AbiNahed[§] *Sarada Dakua*[§] *Georges Younes*[§] *Robin Richardson*[‖] *Peter V. Coveney*[‖]

[⋆] College of Engineering, Qatar University, Doha, Qatar
[†] School of CS and EE, University of Essex, Colchester, UK
[‡] Department of Computer Science, University of Huddersfield, Huddersfield, UK
[§] Department of Surgery, Hamad Medical Corporation, Doha, Qatar
[‖] Centre for Computational Science, University College London, London, UK

ABSTRACT

A weakness in the wall of a cerebral artery will cause a dilation or blood vessel ballooning which is called a cerebral aneurysm. The best treatment for the patients needs fast and accurate diagnosis of the aneurysm. HemeLB is a fluid flow simulation environment in complex geometries developed to help neurosurgeons with the aneurysm blood flow related information, therefore, a cost efficient platform for HemeLB implementation can be employed in hospitals to provide surgeons with the simulation results in real-time. In this work, an improved version of HemeLB for GPU implementation and result visualization is developed. A graphical user interface for smooth interaction with end users is also presented. Finally, a comprehensive evaluation of this implementation is reported. The obtained results demonstrate that the proposed implementation achieves a maximum performance of 15,168,964 site updates per second and is capable to speed up HemeLB for deployment in hospitals and clinical investigations.

***Index Terms*—** Cerebral aneurysm, HemeLB, Visualization, GPU

1. INTRODUCTION

Cerebral aneurysm is an abnormal focal dilation of a brain artery caused by a weakness in blood vessel wall [1]. The number of patients who suffer from cerebrovascular disorders like cerebral aneurysm is growing in non-developed countries [2]. A quick review of the statistics shows that people who already have or will develop brain aneurysm are between 1.5% to 5% of the general population [3]. Another study shows that about 5% of patients with growing cerebral aneurysm and 0.5-1.1% of those with non-growing brain aneurysms will suffer from rupturing [4].

This paper was made possible by National Priorities Research Program (NPRP) grant No. 5-792-2-328 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

An effective treatment of aneurysms is endovascular approaches which reduce the operative risks, pains and cost of hospitalization [5]. These methods which take advantages of intra-aneurysmal coils, may fail due to lack of information about the aneurysm status. In recent years, combination of coils with stents has been widely used as an efficient treatment which reorganizes the blood flow into and around the aneurysm [6]. In order to get the best treatment, the interventional radiologist has to identify the vascular geometry and estimate cerebral blood flow behavior such as pressure and velocity from 2D and/or 3D images. Presently, there are few approaches to measure these flows and related data intraoperatively. Consequently, diagnosis and treatment of aneurysms are highly dependent on the experience and skill of the radiologist.

Modeling and simulation of fluid flow and hemodynamics can provide clinicians with more precise analysis and thus diagnosis of aneurysm. Simulation of fluid flow in large scale and complex geometries requires physical model, substantial computing resources as well as efficient software. Recently, Lattice-Boltzmann (LB) method is widely used for simulation of fluid flows. LB was developed for time-dependent simulation of large systems by using a parallel implementation [7]. In [8], the efficient hardware architecture of LB method is introduced. HemeLB (Hemodynamic lattice-Boltzmann) is a massively parallel LB fluid solver developed to simulate fluid flows in sparse and complex systems on large supercomputing resources [7, 9, 10]. The aim of designing HemeLB was to provide neurosurgeons with timely and clinically relevant assistance [11]. HemeLB blood flow simulation and its performance have been deployed on High Performance Computing (HPC) platforms including HECToR, Blue Waters, SuperMUC and ARCHER [9, 10, 11]. It is required to optimize HemeLB on dedicated computational infrastructures in order to employ the software in clinical environments. Multicore General Purpose Graphical Processing Units (GPGPUs) and Field Programmable Gate Arrays (FPGAs) are probably todays most powerful computational hardware in various ap-

plications such as machine learning and artificial intelligence [12, 13].

The objective of this paper is to develop and evaluate a visualized version of HemeLB on a workstation with CUDA (Compute Unified Device Architecture) capable GPUs. To achieve this, an optimization of the original HemeLB for GPU implementation is introduced. In order to evaluate the proposed implementation, a set of comprehensive tests using real patients data has been carried out on Exxact Tensor workstation with four NVIDIA GPUs.

The rest of the paper is organized as follows. In Section 2, HemeLB lattice-Boltzmann model is described. Section 3 presents HemeLB visualization. The experimental results and their evaluations are reported in Section 4 followed by the conclusions in Section 5.

2. HEMELB MODEL

LB is a fast technique for simulation of large and complex fluid systems by means of parallel implementations [7]. HemeLB code applies parallelized LB method as a fluid solver, which represents fluid with particles. In this method it is assumed that the propagation and collision processes of particles are done over a discrete lattice mesh. In the proposed system, three dimensional lattice with 19 discrete velocity directions (D3Q19) is used. Figure 1 represents the D3Q19 model.

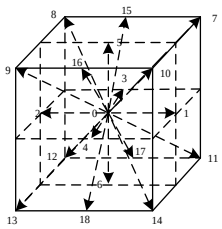


Fig. 1: Lattice node of D3Q19 model [**?**].

HemeLB is implemented with different Boundary Conditions (BC) [7], such as Ladd iolets for velocity inlet BC [14] and Bouzidi-Firdaouss-Lallemand (BFL) for the interpolated wall collision BC [15]. By the use of topology-aware two-level domain decomposition, HemeLB provides a good workload distribution for parallel implementation. In addition, the improvements made in HemeLB decrease redundant operations, improve pattern regularity and enhance intra-machine communications [9].

3. HEMELB IMPLEMENTATION

The benefits of GPU-oriented solutions such as processing speed and large datasets analyzing are achieved by the use of thousands of processor cores in a single chip [12]. In the following section, the real-time visualization of HemeLB on Exxact Tensor workstation with four CUDA capable GPUs is discussed.

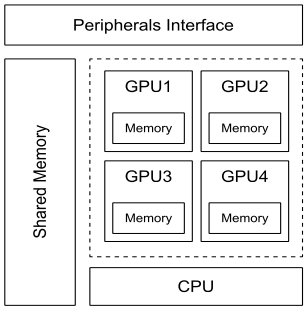


Fig. 2: Simplified architecture of Exxact Tensor TWS-289059-DPN workstation.

3.1. Architecture of Exxact Tensor workstation

The proposed work is implemented on Exxact Tensor TWS-289059-DPN workstation which is known as deep learning NVIDIA GPU solution [16]. The platform consists of one Intel core i7-5960X processor, 4 TB HDD, 1 TB SSD as well as four GeForce GTX 1080 Ti GPUs. Each NVIDIA GPU benefits of 11 GB GDDR5X random access memory with 484 GB/s memory bandwidth and 3584 CUDA cores. The GPU base clock speed is 1481 MHz and the memory is running at 1376 MHz. Figure 2 shows the simplified architecture of the workstation with four NVIDIA GPUs.

3.2. HemeLB configuration

The original version of HemeLB code is available in [17]. The *openmpi* package and CUDA platform are required for the implementation of the proposed system. Once the package is built, compilation flags given in Table 1 are needed to be set as boundary conditions. Table 2 represents the simulation, geometry and inlets/outlets settings used in the HemeLB.

Table 1: HemeLB boundary settings.

cmake settings	Values
HEMELB_INLET_BOUNDARY	LADDIOLET
HEMELB_WALL_INLET_BOUNDARY	LADDIOLETBFL
HEMELB_USE_VELOCITY_WEIGHTS_FILE	ON
HEMELB_WALL_BOUNDARY	BFL

3.3. HemeLB visualization

3.3.1. HemeLB interface

The original HemeLB was designed for command line professional users. In order to enable clinicians to run HemeLB

Table 2: HemeLB simulation, geometry and inlets/outlets settings.

HemeLB settings	Values
Step length	2×10^{-5} s
Voxel size	100×10^{-6}
Inlet velocity	file
Inlet radius	1.9×10^{-3}
Outlet amplitude	0.0 mmHg
Outlet phase	0.0 rad
Outlet period	1.0 s

conveniently, a Graphical User Interface (GUI) has been developed. It provides end users with a smooth interaction environment for running the software. It allows clinicians and neurosurgeons to steer the visualization and manage the simulation operations. Figure 3 illustrates the main GUI window designed with Qt cross-platform framework.

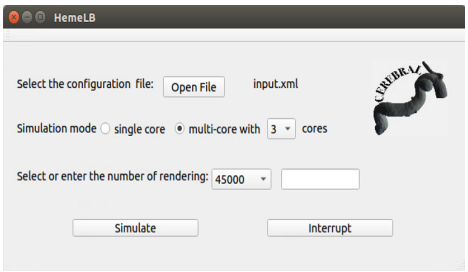


Fig. 3: Main GUI window designed for HemeLB visualization.

The GUI helps clinicians and neurosurgeons to launch HemeLB by a simple click. First, the user should upload the configuration file of the subject under examination, then set some input settings such as simulation mode and rendering number. This GUI gives the control of the simulation mode to the clinicians. The user can choose single core or multicore simulation mode before running the software. The number of cores in multicore mode can also be selected by the end user. Another option is the number of rendering which can be chosen from a list or entered by the clinician. Finally, the *simulate* and *interrupt* buttons give the control of simulation operation to the user. The flowchart of the designed GUI is depicted in Figure 4.

3.3.2. Real time visualization

Visualization of HemeLB is done on devoted CUDA capable GPUs. The LB nodes and the visualization client communicate by the existing message passing interface in HemeLB. Figure 5 shows the architecture of the proposed framework. In Figure 5, Lattice properties are computed and cashed in each compute node then transferred to $RANK0$ node. This node schedules lattice data transfer from the compute nodes

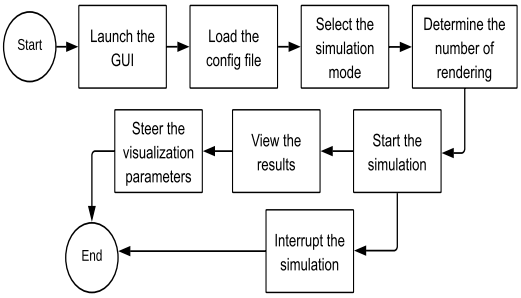


Fig. 4: Flowchart of HemeLB GUI.

and handles the view steering. The incoming lattice data is shared with GPU nodes and rendering is performed at the same time.

The three-tier architecture of the visualization client is presented in Figure 6. The first tier is OpenGL application and the middle one is the host layer which stores the application data such as steering parameters and cached lattice properties. Eventually, all the voxels for direct volume rendering are stored in the rendering layer which is depicted as the top tier. The CUDA cores in the rendering layer perform the volume rendering kernel based on the ray marching algorithm [18].

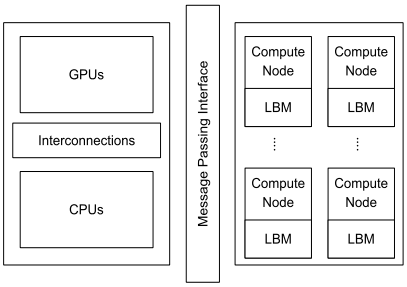


Fig. 5: Architecture of proposed visualization framework.

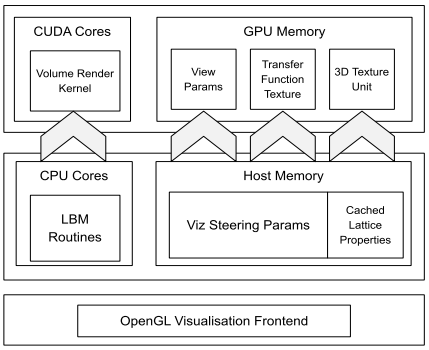


Fig. 6: The three-tier internal architecture of visualization client.

It is well known that the number of memory transfers is critical in real-time applications. In HemeLB visualization, the memory transfers include the transmission of lattice data to the visualization volume and a number of steering simulation. In order to optimize the memory transfers in the proposed framework, the following two-level memory access solution is deployed. In the first place, the visualization volume is stored in a 3D texture unit to enable the fast voxel values sampling. Secondly, in order to store the viewing parameters and the look-up table, the GPU constant memory buffers are used. This allows Direct Memory Access (DMA) which performance is comparable to reading from registers.

The steerable parameters introduced in the visualized HemeLB include zooming, model rotation and adjustment of scaling and offset of the transfer functions.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Five subjects of 3D Rotational Angiography (3DRA) selected by Hamad Medical Cooperation (HMC) clinicians were used to evaluate the proposed solution. Figure 7 represents two STereoLithograph (STL) files used to evaluate HemeLB visualization. In Table 3 the number of lattice sites and blocks for each subject are presented. HemeLB was employed using D3Q19 lattice node model with simple bounce-back boundary condition and a fixed physical viscosity of 0.004 Pa.s. The inlet velocities applied to all STLs in the simulation is illustrated in Figure 8.

(a) Subject 16 (b) Subject 23

Fig. 7: STL files used for evaluation of visualized HemeLB.

Table 3: Overview of simulation domains.

Name	Number of lattice sites	Number of blocks
Subject 16	296,814	15,000
Subject 23	132,910	5,320

Table 4 shows the timing performance of simulating two subjects on the Exxact Tensor workstation. The maximum performance of 15,168,964 Site Updates Per Second (SUPS) is achieved in this evaluation. As it is depicted in Table 4, the larger geometry has the highest SUPS, although the timing performance for different subjects is similar. This insures that the SUPS per core is independent of other factors. The simulation results of two geometries obtained from the proposed version of HemeLB are illustrated in Figure 9.

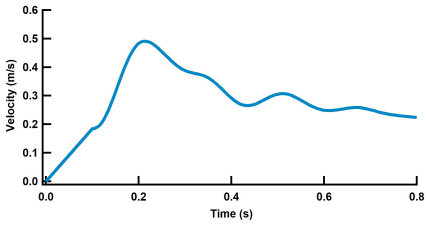


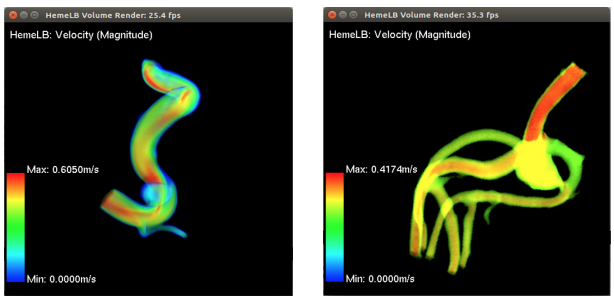
Fig. 8: The inlet blood velocity applied to STLs.

Table 4: Simulation timing data.

Name	Total time (s)	Number of steps	SUPS
Subject 16	906	46,302	15,168,964
Subject 23	297	30,558	13,674,962

5. CONCLUSION

In this paper, a solution for designing and implementing a real-time visualization version of HemeLB on a workstation is presented. The proposed implementation enables HemeLB to be exploited locally in hospitals rather than a distributed environment. In addition, a GUI for HemeLB environment is also introduced in this work. The GUI allows clinicians to launch the HemeLB, view and steer the visualized results in a user friendly and smooth manner. The reported results in this experience, obtained from comprehensive tests performed on real patient data. The results achieve a maximum performance of 15,168,964 site updates per second and demonstrate that the proposed implementation is capable to support HemeLB in clinical environments. A working system which can be used in hospitals for training and practicing the cerebral surgeries can be developed as a future work.



(a) Subject 16 (b) Subject 23

Fig. 9: Results of HemeLB visualization framework.

6. REFERENCES

[1] R.T. Higashida, "What you should know about cerebral aneurysms," *Pamphlet. American Heart Association Cardiovascular Council*, 2003.

[2] N. Benkirane, A. Bennis, S. Bellakhdar, R. Habbal, S. Aidi, M. El Alaoui Faris, H. El Otmani, B. El Moutawakil, M.A. Rafai, and I. Slassi, "Stroke risk factors in a moroccan population: a multicentric prospective study," *Journal of the Neurological Sciences*, vol. 357, pp. e367, October 2015.

[3] T. Mashiko, K. Otani, R. Kawano, T. Konno, N. Kaneko, Y. Ito, and E. Watanabe, "Development of three-dimensional hollow elastic model for cerebral aneurysm clipping simulation enabling rapid and low cost prototyping," *World Neurosurgery*, vol. 83, pp. 351–361, March 2015.

[4] Verywell health, "Brain aneurysm prognosis," https://www.verywellhealth.com/brain-aneurysm-prognosis-3146347, 2018, [Online; accessed 04 Jan. 2019].

[5] A. Molyneux, R. Kerr, International Subarachnoid Aneurysm Trial (ISAT) Collaborative Group, et al., "International subarachnoid aneurysm trial (isat) of neurosurgical clipping versus endovascular coiling in 2143 patients with ruptured intracranial aneurysms: a randomised trial," *Journal of stroke and cerebrovascular diseases*, vol. 11, no. 6, pp. 304–314, October 2002.

[6] B.M. Kim, D.J. Kim, and D.I. Kim, "Stent application for the treatment of cerebral aneurysms," *Neurointervention*, vol. 6, pp. 53–70, August 2011.

[7] M.D. Mazzeo and P.V. Coveney, "HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries," *Computer Physics Communications*, vol. 178, pp. 894914, June 2008.

[8] X. Zhai, A. Amira, F. Bensaali, A. Al-Shibani, A. Al-Nassr, A. El-Sayed, M. Eslami, S. Dakua, and J. Abinahed, "Zynq SoC based acceleration of the lattice-Boltzmann method," *Concurrency and Computation: Practice and Experience*, in press.

[9] D. Groen, J. Hetherington, H.B. Carver, R.W. Nash, M.O. Bernabeu, and P.V. Coveney, "Analysing and modelling the performance of the HemeLB lattice-Boltzmann simulation environment," *Journal of Computational Science*, vol. 4, pp. 412–422, September 2013.

[10] A. Patronis, R.A. Richardson, S. Schmieschek, B.J.N Wylie, R.W. Nash, and P.V. Coveney, "Modelling patient-specific magnetic drug targeting within the intracranial vasculature," *Frontiers in physiology*, vol. 9, pp. 331, April 2018.

[11] D. Groen, R.A. Richardson, R. Coy, U.D. Schiller, H. Chandrashekar, F. Robertson, and P.V. Coveney, "Validation of patient-specific cerebral blood flow simulation using transcranial doppler measurements," *Frontiers in Physiology*, vol. 9, pp. 721, June 2018.

[12] M.A. Suchard, Q. Wang, C. Chan, J. Frelinger, A. Cron, and M. West, "Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures," *Journal of computational and graphical statistics*, vol. 19, no. 2, pp. 419–438, January 2010.

[13] X. Zhai, M. Eslami, E.S. Hussein, M.S. Filali, S.T. Shalaby, A. Amira, F. Bensaali, S. Dakua, J. Abinahed, A. Al-Ansari, et al., "Real-time automated image segmentation technique for cerebral aneurysm on reconfigurable system-on-chip," *Journal of Computational Science*, vol. 27, pp. 35–45, July 2018.

[14] A.J.C. Ladd, "Numerical simulations of particulate suspensions via a discretized Boltzmann equation. part 1. theoretical foundation," *Journal of fluid mechanics*, vol. 271, pp. 285–309, July 1994.

[15] M. Bouzidi, M. Firdaouss, and P. Lallemand, "Momentum transfer of a lattice-Boltzmann fluid with boundaries," *Physics of fluids*, vol. 13, pp. 3452–3459, November 2001.

[16] Exxact Corporation, "Exxact Tensor TWS-289059-DPN specification," https://www.exxactcorp.com/Exxact-TWS-289059-DPN-E289059, 2018, [Online; accessed 25 Dec. 2018].

[17] UCL, "HemeLB Github repository," https://github.com/UCL/HemeLB, 2018, [Online; accessed 10 Oct. 2018].

[18] K. Zhou, Z. Ren, S. Lin, H. Bao, B. Guo, and H.Y. Shum, "Real-time smoke rendering using compensated ray marching," in *ACM Transactions on Graphics (TOG)*. ACM, August 2008, vol. 27, p. 36.